Abstract
In this paper we introduce a novel real-time method to track weakly textured planar objects and to simultaneously estimate their 3D pose. The basic idea is to adapt the classic tracking-by-detection approach, which seeks for the object to be tracked independently in each frame, for non-textured objects. In order to robustly estimate the 3D pose of such objects in each frame, we have to tackle three demanding problems. First, we need to find a stable representation of the object which is discriminative against the background and highly repetitive under severe perspective transformations. Second, we have to relocate this representation in every frame, even under considerable viewpoint changes. Finally, we have to estimate the pose from a single, closed object contour. Of course, all demands shall be accommodated at low computational costs and in real-time.

To attack the above mentioned problems, we propose to exploit the properties of Maximally Stable Extremal Regions (MSERs) for detecting the required contours in an efficient manner and use random ferns as efficient and robust classifier for tracking. To estimate the 3D pose, we construct a perspectively invariant frame on the closed contour which is intrinsically provided by the extracted MSER. In our experiments we obtain robust tracking results with accurate poses on various challenging image sequences at a single requirement: One MSER used for tracking has to have at least one concavity that sufficiently deviates from its convex hull.

Keywords: Planar Tracking, 3D Pose, Tracking-by-Detection

Index Terms: K.6.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking

1 Introduction
Tracking and 3D pose estimation of planar objects is a well investigated area in the field of Augmented Reality (AR) applications. In AR the estimated 3D pose of the planar object is frequently used to superpose computer-generated images over real world views. There are mainly two different directions of research in this field: using fiducial markers or to track natural features. In general, using markers enables robust and accurate tracking results, but in contrast natural feature tracker provide a much more natural experience. Therefore, feature-point based tracking and pose estimation methods were more popular in the last years.

Feature-point based methods mainly differ in the type of features and descriptors used for matching the object model to the current frame. It is important to provide reliable and repeatable feature locations and to describe them in a discriminative way. Thus, the most successful feature-point based tracking approaches are directly based on state-of-the-art feature point detectors and descriptors. For example, Wagner et al. [23] proposed a tracker based on the well-known SIFT descriptor [15]. Their main focus was on an efficient implementation allowing real-time 6-DoF pose estimations even on mobile phones. He et al. [10] introduced a tracker using SURF features [2], which exploit the highly efficient integral image structure enabling fast object tracking. Since feature point description is mostly the performance bottleneck of these methods Özüysal et al. [20] proposed a different approach to overcome this limitation. They showed that in combination with a strong classifier, also most primitive features like simple differences between grayscale values at randomly chosen locations enable accurate and efficient tracking results.

The main problem of all of these feature-point based trackers is that in many applications only little or no texture is present on the object. For example, augmented reality would strongly benefit from replacing the required fiducial markers of predefined shape having a unique pattern, by much more simple, weakly textured objects e.g. a simple shape sketch drawn on the hand as it is illustrated in Figure 1(a). In such cases, almost all feature point based methods would fail, since not enough features are found to get a robust estimate of the current object pose. In contrast, classical template-matching methods can deal with the issue of weak texture. Template matching [1, 7, 12] implies comparison to strong edge priors and is thus well-suited for detection of weakly textured objects. However, for real-time 3D pose estimation template matching mostly becomes impractical due to its computational complexity. Recently, in [4] a direct method using a mutual information based metric for aligning the current image to a reference template was proposed which yields real-time performance and high accuracy. Nevertheless, this approach is likely to fail for tracking weakly-textured objects as considered in our method, because they lack of characteristic pixel intensity distributions, which in turn are exploited in mutual information based methods.

In this paper, we propose a novel approach to track and simultaneously estimate the pose of weakly textured planar objects. Our work is most closely related to the recent tracking approaches proposed in [13] and [9], which are considered as state-of-the-art in the fields of computer vision and augmented reality, respectively.

In [13], Holzer et al. tried to overcome the weaknesses of classical template matching by learning distance transformed templates (DTT) and their spatial relations at specific poses. Our method improves several issues of this method, which we consider as major drawbacks. First, [13] requires extraction of closed contours from Canny edges, which can be hindered by large perspective distortions, changing lighting conditions and reflections. In order to close possible gaps they propagate certain, closed distance transform level curves to the extracted edges, where the choice of this level curve is non-trivial. Second, edge detection by Canny and the error-prone post-processing represent a strong computational bottleneck of the entire method. Finally, object pose can only be estimated from multiple contours, where their spatial relationships have to be learned offline. We consider this as strong restriction since many objects can be defined by a single, closed contour without the availability of adjacent parts.

* email: {donoser, kontschieder, bischof}@icg.tugraz.at
In [9], Hagbi et al. introduced Nestor, a method for real-time recognition and camera pose estimation from planar shapes which runs in real-time on mobile phones. Nestor requires a pre-trained shape library containing shapes to be recognized during tracking. Our method improves over several points of [9]. For contour detection we apply a significantly more robust approach which outperforms the error-prone, simple adaptive thresholding in [9]. Adaptive thresholding assumes that the contrast to the background for all regions to be tracked is quite similar, whereas our approach is able to track differently colored and shaped regions potentially having extremely different boundary contrast. Furthermore, we propose to use a robust learning stage for region assignment, whereas in [9] only simple nearest neighbor analysis on weak signatures is performed. Such an approach will frequently fail in cluttered scenarios, whereas our proposed method achieves the same efficiency and quality even in severely cluttered sequences.

To sum up, our method has several contributions circumventing most of the aforementioned critical parts of [13] and [9]. We propose to use the Maximally Stable Extremal Region (MSER) detector [16] which provides highly repeatable, closed contours independent of the viewpoint. MSER detection is fast and linear time algorithms have been proposed to find them. We further show that the distance transformations of the obtained MSERs can be effectively and efficiently trained in a discriminative manner using a state-of-the-art machine learning method. During test time MSERs detected in the current frame are classified as belonging to our model or to the background in short computation time. We further do not exploit spatial relations between the regions, because we want to be able to track objects consisting of only a single MSER. Therefore, we describe a method for estimating the pose from a single binary region by constructing a frame which is invariant under perspective transformations and delivers accurate and robust pose estimation results. In such a way, we present a lightweight and real-time capable tracking-by-detection framework, with only one limitation. For pose estimation from a single MSER we need at least one concavity that sufficiently deviates from its convex hull.

The outline of the paper is as follows. In Section 2 we describe our planar tracking and pose estimation method in detail. Our method is a model-based approach and hence requires an offline training stage. During test time efficient classification allows robust pose estimation per frame. Section 3 evaluates the properties of our proposed method in detail, illustrating results and low reprojection errors on several sequences, e.g. outperforming publicly available state-of-the-art trackers. Furthermore, we demonstrate robust tracking results and pose estimation results even for a single closed contour e.g. by tracking a sketch manually drawn on a hand. Finally, Section 4 concludes our paper.

2. PLANAR TRACKING AND POSE ESTIMATION

The goal of our method is to robustly track a planar, weakly textured object through a video sequence and to concurrently estimate its 3D pose. Our method requires an offline training step where a template image of the object-to-be-tracked has to be provided. The template must consist of a set of closed contours which are detected by applying the Maximally Stable Extremal Region (MSER) detector on the provided template image. This detection step is also applied during test time on every frame of the video. Therefore, MSER detection is a key part for our robust tracking method and is described in detail in Section 2.1.

To classify detected MSERs, we learn an efficient classifier for assigning MSERs to our training model. Training is described in detail in Section 2.2. During test time we detect MSERs in the image and use the learned classifier to assign each MSER to one of the previously learned classes. We build a perspective frame for all MSERs not assigned to the background class. In Section 2.3 we show how to estimate the pose using the obtained perspective frames which allows to augment the videos with objects like the OpenGL teapot for visualizing the quality of pose estimation. Figure 1 shows some results for different planar objects using e.g. a single detected MSER on a manually drawn sketch on the hand.

2.1 Maximally Stable Extremal Regions

The key part for tracking weakly textured planar objects is to robustly detect closed contours in order to define patches that can be used in the efficient random fern classifier for matching. Closed contour detection has to be applied to the template image before training and additionally to all input images at runtime. Therefore, detection has to be as efficient as possible.

As also outlined by Holzer et al. [13] this is a very challenging step, since perspective distortions and possible edge discontinuities occur frequently due to bad lighting conditions or reflections. Holzer et al. used the results of a Canny edge detector [3] for identifying closed contours. Of course, due to varying lighting conditions, locally weak gradient information and small gaps, often no closed and connected edge pixels sequences are found without further processing. Holzer et al. used distance transformation and level sets at specific manually chosen distances to close these gaps. Nevertheless such a post-processing step is error-prone and requires manual selection of parameters. Furthermore, the Canny edge detection step and the post-processing are the computationally most demanding steps of their entire framework.

We propose to use Maximally Stable Extremal Regions (MSERs), which were introduced by Matas et al. [16], as efficient and effective variant to retrieve closed contours from images. This makes edge detection and post-processing unnecessary. The MSER
detector is one of the best interest region detectors in the field of computer vision. The evaluations in [17] show that MSERs are detecting stable features of arbitrary shape and scale outperforming related detectors. Despite the number of detected regions being low, the repeatability is better than with other detectors (especially for viewpoint changes, as it is essential in an augmented reality scenario) and their detection efficiency is unbeaten. In general MSER detection is applied on a grayscale image and returns a set of connected regions, that are defined by an extremal property of the intensity function in the region and on its outer boundary. MSERs have several properties that form their superior performance as most repeatable local detector. The set of MSERs is closed under continuous geometric transformations and is invariant to affine intensity changes. MSERs are furthermore inherently scale-invariant and no smoothing or additional post-processing steps are required to simultaneously detect coarse and fine structures. Despite this fact, Forssen et al. [6] also showed that a scale-space approach leads to further scale-invariance and higher repeatability.

MSERs are also considered as the most efficient interest region detector, for example Nister and Stewenius [19] recently proposed an algorithm which detects MSERs in linear time. Donoser and Bischof [5] presented a tracking variant of MSER detection which further improves repeatability, but this extension has not been considered in this work. All these properties make MSERs perfectly suited for usage in a planar, weakly textured object tracking framework. Typical MSER detection results are shown in the middle column of Figure 3.

### 2.2 Training an efficient classifier

Our proposed tracking method is a model-based approach and hence requires a training step before it can be applied. We assume that we have given a single ortho-image of the planar object as training data. As a first step we detect MSERs in this training image, which provides the regions of the reference model $\mathcal{F}$. The task of training is to learn the representations of the individual reference MSERs, to be able to classify during runtime an arbitrary detected MSER independent of the current viewpoint in an efficient and robust manner.

For classification we have chosen the randomized ferns method proposed by Özüysal et al. [20], because it is implicitly multi-class capable and has shown to be impressively efficient. The random fern is a semi-naive Bayesian classifier and an adaptation of the random forest concept introduced by Lepeit et al. [14]. The underlying idea is to build a large number of random decision trees, where each node represents a decision that narrows down the choices to a final decision outcome. Random ferns model the conditional probabilities derived from a large number of binary decisions based on simple single feature comparisons. Similar to random forests they can handle high dimensional datasets and do not suffer from overfitting.

We train random ferns using the distance transform representation of the regions in our provided reference model $\mathcal{F}$. Using distance transformations instead of binary region representations has shown to provide more robust results during recognition [13]. All MSERs are scaled to a canonical frame to allow pixel-based node decisions. The total number of classes $K$ is determined by the number of different contours $|\mathcal{F}|$, i.e. the number of detected MSERs, and an additional class for background objects, thus $K = |\mathcal{F}| + 1$. The background class is needed, since the minimum cardinality of $\mathcal{F}$ is one when only a single contour is available. As usual in random forest and fern classification, we cover the possible viewpoints for training by synthetically generating samples from different perspectives on the input model. Training samples for the background class are extracted from MSERs which are detected in arbitrarily chosen input images or from a video stream not containing the object-to-be-tracked.

Figure 2: Perspective frame construction for contour with single concavity.

For training, we employ pixel-based node tests on each input sample $S$, which return a binary decision defined by

$$
\begin{align}
    f(S) &= \begin{cases} 
        1 & \text{if } S(x_1) < (S(x_2) + \tau) \\
        0 & \text{otherwise}
    \end{cases},
\end{align}
$$

Here, $x_1, x_2$ denote the randomly chosen pixel locations and $\tau$ is a threshold variable. Please note, that in contrast to other tracking approaches using random ferns as classifier [11, 13], we do not learn the pose during training. Pose estimation is exclusively based on estimating a perspective frame for each classified region as it is described in Section 2.3.

During runtime, in every frame the set of detected MSERs $\{R_1, \ldots, R_M\}$ is classified to one of the previously trained classes $c_1, \ldots, c_K$ after mapping to the canonical representation. The class labels $c(R_i)$, $i = 1, \ldots, M$ are assigned according to

$$
c(R_i) = \underset{k}{\text{argmax}}\ P(c_k|R_i),
$$

where $P(c_k|R_i)$ is the product of class conditional probabilities returned from the random ferns. After discarding all regions that are classified as background, we denote $\mathcal{F}$ as the resulting set of regions with assigned class labels, which are considered for pose estimation.

### 2.3 Robust pose estimation

After classification of all MSERs in the current frame providing the candidate set $\mathcal{F}$, we estimate a homography and the pose for augmenting our videos. To estimate the homography for the classified MSERs in $\mathcal{F}$, we need to find at least four proper point correspondences to their respective model templates in $\mathcal{F}$.

While many techniques exist for estimating homographies based on interest point matches between images, estimation from binary images or shapes received less attention. One approach was recently proposed by Nemeth et al. [18], who estimated a homography between two shapes by solving a system of nonlinear equations generated by integrating linearly independent functions over the domains determined by the shapes. Unfortunately, registration requires a few seconds per match, which of course is not applicable in the tracking domain.

In general it is not possible to uniquely determine the 3D pose from convex (e.g. circular or rectangular), planar contours under perspective transformations. Therefore, we require at least one concavity on the extracted MSER contours as indicated in Figure 2 and discard all regions failing this concavity test in the first place. To eliminate rough outlier regions with a single concavity, we assume locally affine conditions and apply an area ratio test that seeks...
for similarities between all regions \( R \in \mathcal{R} \) and their corresponding model templates \( T \in \mathcal{T} \) in the areas under the convex hull \( A_H(.) \) and the contours \( A_C(.) \), respectively. We formulate this area ratio test as
\[
\left| \frac{A_H(T)}{A_C(T)} - \frac{A_H(R)}{A_C(R)} \right| \leq \gamma ,
\]
where \( \gamma \) is a threshold that controls the maximally allowed deviation from the template area ratio.

To account for multiple concavities, this area test can easily be extended as follows: Let \( m \) be the number of significant concavities available for a region \( R \) and its corresponding model template \( T \). Furthermore, let \( A_{D_j}(.) \) be a function providing the area produced by the convex hull for concavity \( k \). Since all concavities on \( R \) and \( T \) appear in a fixed and ordered manner, we can define a cost function \( S(.) \); given a pair of concavity indices \( i,j \) as
\[
S(T,R|i,j) = \sum_{k=0}^{m-1} \left| \frac{A_H(T) \setminus A_{D_j}(T)}{A_C(T)} - \frac{A_H(R) \setminus A_{D_j}(R)}{A_C(R)} \right| ,
\]
with
\[
i' = (i+k) \mod m
\]
\[
j' = (j+k) \mod m.
\]

The minimal cost for a chosen concavity index \( i \) on \( T \) can be obtained as offset \( v \) between the enumerating indices and similarly to (3) used for outlier removal as follows:
\[
\left\{ \min_{v \in [0,..,m-1]} \left( S(T,R|i,v+1) \right) \right\} \leq \gamma .
\]

On the remaining candidate regions we construct the perspective invariant frame according to the method of Ruiz et al. in [21]. For simplicity reasons, we consider a contour with only a single concavity \( r \) as illustrated in Figure 2. Please note that for optimal construction of the frame, the subcontour defining \( r \) should be acute-angled in order to unambiguously deliver the desired correspondences for subsequent homography computation. However, \( r \) is not directly needed for homography estimation but instead serves in the frame construction as an accuracy indicator point where diagonals \( \overline{cd} \) and \( \overline{kq} \) should finally intersect.

The frame construction starts from an arbitrarily chosen point \( k \) on the bitangent of \( r \). Second, we draw a tangent through \( r \), delivering two intersections \( a \) and \( b \) with the convex hull. A pre-fixed cross ratio determines the location of point \( q \). In the next step, we construct the tangents from \( q \) to both sides of the convex hull and gather point \( c \) in the intersection with the bitangent of \( r \). The last point \( d \) is the intersection of the tangents drawn from \( k \) and \( q \) to the convex hull, respectively. Depending on the choice of \( k \), the diagonals \( \overline{cd} \) and \( \overline{kq} \) closely intersect at \( r \), which is required for obtaining accurate point correspondences. This procedure is applied once to all templates in \( \mathcal{F} \) and to all candidate region contours during runtime. In case of multiple concavities on a single contour, the perspective frame is individually constructed for each concavity, whereas matching correspondences are obtained as a by-product from (5).

From the resulting correspondences, we estimate the homography \( H \) for a region \( R \) using the standard direct linear transform. As a final acceptance test, we use the resulting homography and calculate the mean reprojection error \( \text{err}(R) \) of the contour points \( C(R) \) into the distance transformed model contour \( D_T \) according to
\[
\text{err}(R) = \frac{1}{|C(R)|} \sum_{x \in C(R)} D_T(H^{-1}(C(R);x)) .
\]

Given the point correspondences defined by the perspective frame of all MSERs we finally estimate the 3D pose from our obtained point correspondences. There is a huge number of different approaches for estimating the pose in real-time but many of them suffer from pose ambiguities. These ambiguities are the reason for the frequently occurring pose jumps in many planar tracking applications. We apply a method proposed by Schweighofer and Pinz [22], which enables unique, efficient and robust pose estimation and has shown to outperform several other methods in this scope.

3 EXPERIMENTAL EVALUATION

In Section 3.1 we describe the configuration of our proposed method and its parameters and introduce the data sets used for experimental evaluation. In Section 3.2 we show that we qualitatively outperform the state-of-the-art tracking algorithms of [20, 8]. For a quantitative analysis, we provide reprojection error curves during significant variations of the perspective viewpoint.

3.1 System configuration and data sets

The necessary parameters for the random ferns classifiers are fixed to the following settings for all our tests: We grow 50 Ferns with a depth of 12 bit and synthetically generate 40,000 training samples from each extracted MSER model template. The size of the canonical training samples is 50 × 50. We implemented our method in C++ and all experiments run on a standard Dual-Core desktop PC with 4GB of RAM in real-time (20fps) on a frame size of 640 × 480.

For the construction of the perspective frame as described in Section 2.3, we point out the importance of the choice for the cross ratio parameter that fixes point \( q \) (see Figure 2). This parameter 'spans' the frame over the contours and significantly influences the perspective coefficients in the resulting homographies. In such a way, it is important to keep point \( q \) close enough to the contour itself which results in cross ratio values between [0.01 ... 0.03]. The parameter used for rough outlier rejection in (3) and (5) was experimentally determined as \( \gamma = 15 \). Finally, we considered a concavity as significant when its normal distance to the convex hull was ≥ 10 pixels.

For quantitative evaluation of our proposed system, we provide three different image sequences which we name 'California' (887 frames), 'Heart' (482 frames) and 'Sketch' (1214 frames). In addition, we provide qualitative detection results on another sequence denoted as 'Greece'. The image sequences are recorded with a standard USB-webcam and suffer from changes in illumination, motion blur, camera-triggered brightness and non-linear distortions, and we additionally covered a broad range of possible perspective transformations. The frame size is 640 × 480 for all sequences. While on the 'California', 'Greece' and 'Heart' sequences we only learn a single extracted MSER contour, we train on three different contours on the 'Sketch' data set. The image used for the 'Sketch' scene is taken from Holzer et al. [13]. Figure 3 illustrates 'Heart' and 'Sketch' sequences, showing a single input frame on the left, the detected and classified MSERs in a color-coded manner in the middle and resulting pose estimations illustrated with the OpenGL teapot on the right.

3.2 Qualitative evaluation

The best evaluation to demonstrate the quality of our method would be to directly compare it to the two most related methods of [9, 13] on the same test sequences. Unfortunately, for both methods no code is publicly available. Nevertheless, as outlined in detail in the introduction we strongly believe that due to the directly comparable improved properties of the individual steps used in our method we will outperform both approaches on most sequences.

Figure 5 illustrates selected frames from the 'Greece' sequence demonstrating the high accuracy of the obtained pose estimation results by augmenting the stream with the OpenGL teapot. Corresponding videos are provided as additional material. Despite clutter, illumination changes and several confusable regions, our method is mostly able to accurately track the target objects. In rare
Figure 3: Exemplary detection and pose estimation results for different model templates. First column shows the input image, second column shows the detected and classified (color-coded) MSERs and third column shows the input image, augmented with the oriented OpenGL teapot. Please note, that in the first row we only use a single contour for pose estimation, whereas in the second row three contours are analyzed which enables handling of strong occlusions.

Figure 4: Sample images from 'Heart' data set with inaccurate and false detection results from two related state-of-the-art tracking by detection methods. First row: [20], Second row: [8]

cases we experienced jittering effects or unstable pose computations which are mostly due to non-exact MSER extraction and a lack of pose coherency checks (e.g. objects will not perform arbitrary large pose jumps between subsequent frames). However, single frame, tracking-by-detection approaches are generally known to suffer from these drawbacks.

We further compare our results to two publicly available state-of-the-art tracking approaches: a random ferns based tracker [20] and an online boosting tracker [8]. The work in [20] learns simple interest points (extrema of the Laplacian over three octaves) with a Random Fern classifier on a frontal view of a given planar target. During runtime, the most probable interest point class is selected and used for homography estimation. In contrast, the work of [8] is an appearance based tracker which employs online-boosting for feature selection analyzing Haar-type features. For [20], we provide the same frontal view (with a bounding box) as used for model template extraction in our method for offline training. The method of [8] only requires initialization of the target model in the first frame. Both methods immediately fail on all of our data sets, where [20] only delivers (non-accurate) detection results, when the learned object is directly in front of the camera. In Figure 4, we show some selected detection results for these two methods.

We explicitly point out the comparison with tracking-by-detection based methods, since they allow to automatically reinitialize the tracker after occlusion. Comparison to region based approaches (e.g. active contours or level sets) will most likely fail in these sequences, because they easily loose track during occlusion and fast camera or object movements and are therefore not suitable for application in Augmented Reality.

3.3 Quantitative evaluation

In order to provide a quantitative analysis of our method, we evaluate the mean reprojection error as defined in Equation (6) on the previously described data sets. Due to the lack of ground truth information, monitoring of this error value is a reasonable approach to estimate the influence of various changes in the viewpoint. Subsequently, the smaller the reprojection error, the higher the robustness against perspective viewpoint changes.

We illustrate reprojection error results for all three analyzed sequences in Figures 6, 7 and 8. As can be seen, the reprojection error does not significantly change throughout the spectrum of possible viewpoint changes. We observe similar behavior on all three sequences, with an average reprojection error of 2.55 for the 'Greece', 1.98 for 'Sketch' and 2.49 for 'Heart' sequence. Please note, that 'Heart' is especially challenging for pose estimation, since the palm of the hand is not fully planar. For 'Greece' and 'Sketch' only a single MSER was used to estimate the pose, whereas for 'Sketch' we consider three contours for correspondence estimation, which allows us to handle partial occlusions.

4 Conclusion

In this paper, we proposed a novel approach for tracking planar objects and estimating their 3D pose. In contrast to existing methods,
our method is able to robustly detect and estimate the pose from objects consisting of only a single closed contour. Given the current frame, we extract Maximally Stable Extremal Regions which provide robust and closed contours in an efficient manner. The extracted MSERs are warped in a normalized frame and then distance transformed, before they are classified by a random fern, which was trained on synthetic views of the provided template image. Matched contours are then used to estimate a perspective frame which allows estimating the 3D pose even from a single contour. The experiments showed that our method outperforms state-of-the-art methods for tracking planar weakly textured targets.

Acknowledgements. We acknowledge financial support of the Austrian Science Fund (FWF) from project 'Fiber-morph' (P22261-N22) and the Research Studios Austria Project µSTRUCSCOP (818651).

REFERENCES

Figure 6: Reprojection error for 'California' scene with exemplary corresponding images. The second row shows the detection and pose estimation results for frames 22, 347, 443 and 574 (see arrows for positions in Figure), whereas the highest error is observed for partial occlusion.

Figure 7: Reprojection error for 'Heart' scene with exemplary corresponding images. The second row shows the detection and pose estimation results for frames 74, 161, 285 and 320 (see arrows for positions in Figure), whereas higher error is observed due to non-exact contour boundary segmentation.

Figure 8: Reprojection error for 'Sketch' scene with exemplary corresponding images. The second row shows the detection and pose estimation results for frames 150, 394, 505 and 897 (see arrows for positions in Figure). The jitter is due to the non-jointly optimized backprojected contours. Please note, that we trained multiple contours and can therefore handle partial occlusions as shown in the last image of the second row.


